Journal of Numerical Error

Vol. 1, no. 1 (2025), 1-13 jne.mathsophical.com
Identifier: https://n2t.net/ark:/13960/s2cz3rw57p4

RESEARCH ARTICLE

Runge-Kutta computation of the error term in Gauss-Legendre quadrature

J.S.C. Prentice *

Senior Research Officer, Mathsophical, Johannesburg, South Africa
*Corresponding author: jpmsro@mathsophical.com

[Received: 4 July 2024, Published: 1 January 2025]

Abstract
We show how the error term for Gauss-Legendre quadrature can be estimated, by solving a suitable initial value problem using a Runge-Kutta method. The error term can then be used as a correction term, yielding a more accurate result for the quadrature. We have also identified a *condition function* that can amplify/suppress the effect of the Runge-Kutta global error.

Keywords: Gauss-Legendre, Runge-Kutta, error, initial value problem, correction term

Mathematics Subject Classification 2020: 65D30, 65L05, 65L70

1 Introduction

We have recently considered the possibility of determining the error term in Trapezium quadrature by solving a suitable initial-value problem [1]. In this paper, we extend this idea to Gauss-Legendre quadrature.

2 Relevant concepts

We establish here notation and terminology relevant to the rest of the paper.

2.1 Gaussian Quadrature

A *Gaussian quadrature rule* [2][3] has the form

$$G[f, w, a, x, n+1] \equiv \sum_{i=0}^{n} c_i^w f\left(x_i^w\right),$$

where the parameters c_i^w and x_i^w denote the *weights* and *nodes*, respectively. A Gaussian rule is intended to approximate the integral

$$I[f,w,a,x] \equiv \int_a^x w(x) f(x)\, dx,$$

where $w(x) \geqslant 0$ on $[0,x]$ is a *weight function.* We necessarily assume that

$$f : \mathbb{R} \longrightarrow \mathbb{R}$$
$$w : \mathbb{R} \longrightarrow \mathbb{R}$$

and that $f(x)$ is as differentiable as it needs to be for our purposes. Naturally, we also assume that $w(x) f(x)$ is integrable on any interval that we consider.

The following statement gives the relationship between the weights, nodes and degree of precision of a Gaussian quadrature rule [4]: The $(n+1)$-point quadrature rule

$$G[f,w,a,x,n+1] \equiv \sum_{i=0}^{n} c_i^w f\left(x_i^w\right) \approx \int_a^x w(x) f(x)\, dx,$$

where $w(x)$ is a weight function on $[a,x]$, has degree of precision $2n+1$ if and only if the nodes $\left\{x_0^w, x_1^w, \ldots, x_n^w\right\}$ are the roots of the w-orthogonal polynomial $P_{n+1}^w(x)$ on $[a,x]$, and the weights c_i^w are given by

$$c_i^w = \int_a^x w(x) \prod_{\substack{j=0\\ j\neq i}}^{n} \left(\frac{x - x_j^w}{x_i - x_j^w}\right) dx.$$

The approximation error in $G[f,w,a,x,n+1]$ is given by

$$\int_a^x w(x) f(x)\, dx - G[f,w,a,x,n+1] = \frac{f^{(2n+2)}(\zeta(x))}{(2n+2)!} \int_a^x w(x) \left(\prod_{i=0}^{n} \left(x - x_i^w\right)\right)^2 dx,$$

where $\zeta \in (a,x)$.

The most usual case that arises in quadrature applications is the case where $w(x) = 1$. In this case, we write

$$G[f,1,a,x,n+1] \equiv \sum_{i=0}^{n} c_i f(x_i) \approx \int_a^x f(x)\, dx$$

$$c_i = \int_a^x \prod_{\substack{j=0\\ j\neq i}}^{n} \left(\frac{x - x_j}{x_i - x_j}\right) dx$$

$$\int_a^x f(x)\, dx - G[f,1,a,x,n+1] = \frac{f^{(2n+2)}(\zeta(x))}{(2n+2)!} \int_a^x \left(\prod_{i=0}^{n} (x - x_i)\right)^2 dx$$

and the nodes x_i are the roots of the *Legendre polynomial* $P_{n+1}(x)$ on $[a, x]$. Gaussian quadrature with the unit weight function is known as *Gauss-Legendre* (GL) quadrature.

For the sake of clarity in the next section, we will replace the integration variable x with t, while the upper bound in the integral remains as x. Also, we will denote the nodes by t_i. So, we have

$$\int_a^x f(t)\,dt - G[f, 1, a, x, n+1] = \frac{f^{(2n+2)}(\zeta(x))}{(2n+2)!}\int_a^x \left(\prod_{i=0}^{n}(t-t_i)\right)^2 dt \tag{1}$$

2.1.1 Gaussian Quadrature in terms of a Stepsize

From this point onwards, we consider the specific case of GL quadrature. The roots of $P_{n+1}(x)$ on $[a, x]$ are symmetrically distributed within $[a, x]$, but they are not equispaced and they are not located at a or x (so GL quadrature is open). Nevertheless, it is possible to write GL quadrature in terms of a stepsize, as can be done for Newton-Cotes quadrature.

We define

$$h \equiv \frac{x-a}{n+2},$$

so that h is the average separation of the nodes on $[a, x]$. We may now write

$$G[f, 1, a, x, n+1] = h\sum_{i=0}^{n} d_i f(x_i),$$

where

$$d_i \equiv \frac{(n+2)\,c_i}{x-a}.$$

Furthermore, with the substitution $t = a + sh$, where $s \in [0, n+2]$ is a continuous variable, and $t_i = a + \sigma_i h$, where σ_i is an appropriate constant, we find

$$\int_a^x \left(\prod_{i=0}^{n}(t-t_i)\right)^2 dx = h^{2n+3}\int_0^{n+2}\left(\prod_{i=0}^{n}(s-\sigma_i)\right)^2 ds,$$

so that

$$\int_a^x f(t)\,dt - G[f, 1, a, x, n+1] = \frac{h^{2n+3} f^{(2n+2)}(\zeta(x))}{(2n+2)!}\int_0^{n+2}\left(\prod_{i=0}^{n}(s-\sigma_i)\right)^2 ds, \tag{2}$$

from which we conclude that GL quadrature is of order $2n+3$.

The nodes and weights for GL quadrature for the interval of integration $[-1, 1]$ are well-documented; a useful source is [5] (the Legendre polynomials can be generated using *Bonnet's recursion* given in the Appendix. Thereafter, the roots can be found using numerical

software). It is easy to transform them to the case of an arbitrary interval $[a, x]$. If $\widetilde{t}_i$ is a node on $[-1, 1]$, then the corresponding node on $[a, x]$ is given by

$$t_i = \left(\widetilde{t}_i + 1\right)\left(\frac{x-a}{2}\right) + a$$

and the relationship between the corresponding weights is given by

$$c_i = \left(\frac{x-a}{2}\right)\widetilde{c}_i,$$

where $\widetilde{c}_i$ is the weight associated with $\widetilde{t}_i$ on $[-1, 1]$. Note that t_i and c_i have now been expressed linearly in terms of x.

2.2 Runge-Kutta methods

We will assume that the reader has sufficient familiarity with *Runge-Kutta* (RK) methods for solving ODEs numerically. The RK method that we will employ here is an explicit seventh-order method, which we denote RK7. We refer the reader to Butcher [6] and Hairer *et al* [7] for information regarding these methods.

3 Theory

We now have

$$G[f, 1, a, x, n+1] \equiv \sum_{i=0}^{n}\left(\frac{x-a}{2}\right)\widetilde{c}_i f\left(\left(\widetilde{t}_i + 1\right)\left(\frac{x-a}{2}\right) + a\right),$$

which we shall denote by $\widetilde{G}_f(a, x, n)$ from now on.

The integral in (2) will be denoted by

$$\Upsilon(n) \equiv \frac{1}{(2n+2)!}\int_0^{n+2}\left(\prod_{i=0}^{n}(s-\sigma_i)\right)^2 ds$$

and is easily determined for any given value of n (see item #1 in the Appendix).

So, (2) becomes

$$\int_a^x f(t)\,dt - \widetilde{G}_f(a, x, n) = h^{2n+3} f^{(2n+2)}(\zeta(x))\,\Upsilon(n).$$

Differentiating with respect to x gives

$$f(x) - \frac{d\widetilde{G}_f}{dx} = \left(f^{(2n+2)}(\zeta)\frac{dh^{2n+3}}{dx} + h^{2n+3}\frac{df^{(2n+2)}(\zeta)}{d\zeta}\frac{d\zeta}{dx}\right)\Upsilon(n),$$

which can be rearranged to yield

$$\begin{aligned}\frac{d\zeta}{dx} &= \frac{\frac{f(x)-\frac{d\widetilde{G}_f}{dx}}{\Upsilon(n)} - f^{(2n+2)}(\zeta)\frac{dh^{2n+3}}{dx}}{h^{2n+3}\frac{df^{(2n+2)}(\zeta)}{d\zeta}} \\ &= \frac{f(x) - \frac{d\widetilde{G}_f}{dx} - \Upsilon(n) f^{(2n+2)}(\zeta)\left(\frac{(2n+3)h^{2n+2}}{n+2}\right)}{\Upsilon(n)\, h^{2n+3} f^{(2n+3)}(\zeta)}.\end{aligned} \tag{3}$$

Solving this ODE on $[0, x]$ with a suitable initial value gives $\zeta(x)$, which then allows the error term to be determined.

4 Implementation

Implementing a RK method to solve (3) requires addressing several issues.

1. The factor $f^{(2n+3)}(\zeta)$ in the denominator could be zero, or uncomfortably close to zero. We solve this problem by introducing a *monomial offset*, as we suggested in [1]. Essentially, we define a new function
$$g(x) \equiv f(x) + \frac{Dx^{2n+3}}{(2n+3)!},$$
where
$$D = 1 + \max_{[0,x]} \left| f^{(2n+3)}(x) \right|,$$
and we apply our algorithm to $g(x)$, instead of $f(x)$. This gives
$$\begin{aligned}\frac{d\zeta_g}{dx} &= \frac{\frac{g(x)-\frac{d\widetilde{G}_g}{dx}}{\Upsilon(n)} - g^{(2n+2)}\left(\zeta_g\right)\frac{dh^{2n+3}}{dx}}{h^{2n+3}\frac{dg^{(2n+2)}\left(\zeta_g\right)}{d\zeta_g}} \\ &= \frac{g(x) - \frac{d\widetilde{G}_g}{dx} - \Upsilon(n) g^{(2n+2)}\left(\zeta_g\right)\left(\frac{(2n+3)h^{2n+2}}{n+2}\right)}{\Upsilon(n)\, h^{2n+3} g^{(2n+3)}\left(\zeta_g\right)},\end{aligned} \tag{4}$$
where ζ_g and $\widetilde{G}_g$ are quantities relevant to the use of $g(x)$ in (2). The result of this computation can be manipulated to yield the error term corresponding to $f(x)$. See item #2 in the Appendix for detail.
2. The factor h^{2n+3} in the denominator could also be problematic for values of x close to a. To counter this, we choose an initial value x_0 reasonably far from a, and then apply the RK method from x_0 up to a chosen upper limit x_N. This gives the error term on the interval $[x_0, x_N]$, but not on the interval $[a, x_0]$. To find the error term on $[a, x_0]$, we apply the RK method from x_0 *down* to a (a "backward" computation, so to speak) This action requires the use of a limit b in the integral (where $x < b \leqslant x_N$), rather than a, and a suitable initial value, which is not the same as the initial value used for the *forward* computation (we discuss these initial values in the next point). We provide some detail regarding this procedure in items #3 and #4 in the Appendix. The nett result is a piecewise construction of the error term over the intervals $[a, x_0]$ and $[x_0, x_N]$.

3. It is necessary to determine an initial value in order to implement the RK method. Substituting x_0 into (1) gives

$$\int_a^{x_0} f(t)\,dt - \widetilde{G}_f(a, x_0, n) = \frac{f^{(2n+2)}(\zeta(x_0))}{(2n+2)!} \int_a^{x_0} \left(\prod_{i=0}^{n} (t - t_i) \right)^2 dt. \tag{5}$$

If we evaluate the integral on the LHS to a suitable accuracy, (5) becomes a nonlinear equation that can be solved (numerically, if necessary) for $\zeta(x_0)$. Of course, the nodes t_i are the relevant GL nodes on $[a, x_0]$. This gives the initial value $(x_0, \zeta(x_0))$ for the forward RK computation on $[x_0, x_N]$. A suitable initial value for the backward RK computation on $[a, x_0]$ can be found by changing the limits of integration in (5) to give

$$\int_{x_0}^{b} f(t)\,dt - \widetilde{G}_f(x_0, b, n) = \frac{f^{(2n+2)}(\zeta(x_0))}{(2n+2)!} \int_{x_0}^{b} \left(\prod_{i=0}^{n} (t - t_i) \right)^2 dt.$$

Here, the upper limit in the integral is now b, the lower limit is x_0, and the nodes t_i are the relevant GL nodes on $[x_0, b]$. The initial value $(x_0, \zeta(x_0))$ obtained here is **not** the same as in the previous case (since the integration limits are different), and it is probably a good idea to distinguish them through the notation

$$(x_0, \zeta_a(x_0)) \text{ and } (x_0, \zeta_b(x_0)).$$

Of course, even if it is necessary to work with $g(x)$, instead of $f(x)$, the same process can be applied.

5 Examples

By way of numerical examples, we consider

$$\int_0^x t^2 \sin t dt = 2x \sin x - \left(x^2 - 2\right) \cos x - 2$$

$$\int_0^x e^t dt = e^x - 1$$

$$\int_0^x \sin t dt = 1 - \cos x$$

$$0 \leqslant x \leqslant 10 \quad (\Rightarrow a = 0, x_N = 10).$$

Higher derivatives of $f(x) = x^2 \sin x$ and $f(x) = \sin x$ have zeroes on $[0, 10]$ so that the use of a monomial offset is justified for these cases. Also, we will use $x_0 = 0.5$ and $b = 1$ for all three examples. For the sake of economy of presentation, we restrict our calculations to GL quadrature with $n = 2$, although our algorithm is general with regard to n.

If $\delta_\zeta(x)$ denotes the global error in $\zeta(x)$ due to the RK computation, we have

$$\begin{aligned} h^{2n+3} f^{(2n+2)}(\zeta(x)+\delta_\zeta(x)) \Upsilon(n) &= h^{2n+3}\Upsilon(n)\left(f^{(2n+2)}(\zeta(x)) + \delta_\zeta(x) f^{(2n+3)}(\zeta(x)) + \ldots\right) \\ &= h^{2n+3}\Upsilon(n) f^{(2n+2)}(\zeta(x)) \\ &\quad + \delta_\zeta(x) h^{2n+3}\Upsilon(n) f^{(2n+3)}(\zeta(x)) + \ldots, \end{aligned}$$

where we see that

$$h^{2n+3}\Upsilon(n) f^{(2n+3)}(\zeta(x))$$

serves as a *conditioning function*, possibly amplifying $|\delta_\zeta(x)|$.

We define $E^m_{0,N}$ as the maximum magnitude of the difference between the true GL error and the GL error computed via $\zeta(x)$ on $[x_0, x_N]$, $E^m_{a,0}$ as the maximum magnitude of the difference between the true GL error and the GL error computed via $\zeta(x)$ on $[a, x_0]$, and $C^m_{0,N}$ as the maximum magnitude of $h^{2n+3}\Upsilon(n) f^{(2n+3)}(\zeta(x))$ on $[x_0, x_N]$.

In Table 1, we show values for various parameters of significance for each example. For D and $C^m_{0,N}$, we have rounded up to the nearest integer. We did not need a monomial offset for $f(x) = e^x$.

Table 1: Values of various parameters of significance, with $n = 2$.

$f(x)$	$\zeta_a(x_0)$	$\zeta_b(x_0)$	D	$E^m_{0,N}$	$E^m_{a,0}$	$C^m_{0,N}$
$x^2 \sin x$	0.2498	0.7493	112	3.6×10^{-10}	3.7×10^{-16}	907
e^x	0.2519	0.7519	–	2.2×10^{-11}	5.2×10^{-16}	1534
$\sin x$	0.2505	0.7510	2	5.3×10^{-12}	2.6×10^{-16}	10

Figures 1 – 3 (see end of paper) pertain to the example $f(x) = x^2 \sin x$. In Figure 1, we show $\zeta(x)$ on $[0, x_0]$ and $[x_0, 10]$. The discontinuity is simply due to the two different initial values $\zeta_a(x_0)$ and $\zeta_b(x_0)$. In Figure 2 we show

$$\int_0^{10} f(t)\,dt - G[f,1,0,10,3]$$

and

$$\int_0^{10} f(t)\,dt - \left(G[f,1,0,10,3] + \frac{h^7 f^{(6)}(\zeta(x))}{6!}\int_0^4 \left(\prod_{i=0}^{2}(s-\sigma_i)\right)^2 ds\right).$$

The first of these is the *uncorrected* GL error, and the second includes the error computed via $\zeta(x)$ as a *correction term*. There is a clear improvement in accuracy.

We observe, however, that both the uncorrected and corrected errors generally increase with x in Figure 2. We believe that there are two reasons for this: (i) the global RK error

in $\zeta(x)$ is likely to increase with x, and (ii) the condition function shows a clear increase in magnitude with x (see Figure 2), which only serves to amplify the effect of $\delta_\zeta(x)$ on the overall result. Indeed, the corrected error in Figure 3 is increased by some three orders of magnitude near the end of the interval due to the value of the condition function.

6 Conclusion

We have determined the error in Gauss-Legendre quadrature by solving a suitable ODE for $\zeta(x)$ - the central parameter in Gauss-Legendre quadrature - using an explicit Runge-Kutta method. Our algorithm is robust in that it caters for singularities and near-singularities in the ODE. Numerical experiments have shown that correcting the Gauss-Legendre approximation using the error determined in this way improves the accuracy of the quadrature by as much as 12 orders of magnitude. Our study has exposed the existence of a conditioning function, which can amplify the Runge-Kutta global error in $\zeta(x)$. This, in turn, suggests the need to control the Runge-Kutta global error in a step-by-step manner, as we have investigated previously [8][9][10]. We reserve this line of inquiry for future research.

We have only considered Gauss-Legendre quadrature here, wherein the weight function $w(x)$ is unity. Using other weight functions may be as simple as replacing $f(x)$ with $w(x)f(x)$ as the function to be integrated. This may necessitate a change of variable, as in the case of the Chebyshev weight function, but this, too, will be a topic of future research.

Appendix

1. The integral $\Upsilon(n)$ is easily determined. We demonstrate for $\Upsilon(2)$. With $n = 2$ we have $n+1 = 3$ nodes. These are the roots of the Legendre polynomial $P_3(x)$ on $[-1,1]$, and are known to be

$$t_i \in \left\{-\sqrt{\frac{3}{5}}, 0, \sqrt{\frac{3}{5}}\right\}.$$

   Using $t_i = -1 + \sigma_i h$ and $h = 1/2$ we find

$$\sigma_i = \frac{t_i + 1}{h} \in \left\{2 - 2\sqrt{\frac{3}{5}}, 2, 2 + 2\sqrt{\frac{3}{5}}\right\}.$$

   Hence,

$$\begin{aligned}\Upsilon(2) &= \frac{1}{6!}\int_0^4 \left(\prod_{i=0}^{2}(s-\sigma_i)\right)^2 ds \\ &= 0.00812698412698047503721632267570 39,\end{aligned}$$

   and similarly for other values of n. As a matter of interest, we also find

$$\Upsilon(3) = 0.0010984263083575790571899677416923.$$

2. The risk of a singularity in (3), due to the possibility of $f^{(2n+3)}(x) = 0$ in the denominator on the RHS, is easily mitigated. Since we know $f^{(2n+3)}(x)$, it is easy to find a constant D such that $f^{(2n+3)}(x) + D$ is not close to zero anywhere on $[a, x]$ for all values of x of interest. Indeed, a suitable choice for D is

$$D = 1 + \max_{[0,x]} \left| f^{(2n+3)}(x) \right|.$$

Define

$$g(x) \equiv f(x) + \frac{Dx^{2n+3}}{(2n+3)!}.$$

Hence,

$$\int_a^x g(t)\,dt = \int_a^x f(t)\,dt + \int_a^x \frac{Dt^{2n+3}}{(2n+3)!}dt$$

and GL applied to $g(x)$ is

$$\begin{aligned} \widetilde{G}_g(a, x, n) &= G\left[f + \frac{Dx^{2n+3}}{(2n+3)!}, 1, a, x, n+1\right] \\ &= \widetilde{G}_f(a, x, n) + G\left[\frac{Dx^{2n+3}}{(2n+3)!}, 1, a, x, n+1\right]. \end{aligned}$$

This gives

$$\begin{aligned} \int_a^x g(t)\,dt - \widetilde{G}_g(a, x, n) = & \int_a^x f(t)\,dt - \widetilde{G}_f(a, x, n) \\ & + \int_a^x \frac{Dt^{2n+3}}{(2n+3)!}dt - G\left[\frac{Dx^{2n+3}}{(2n+3)!}, 1, a, x, n+1\right], \end{aligned}$$

so that

$$\begin{aligned} \int_a^x f(t)\,dt - \widetilde{G}_f(a, x, n) = & \left(\int_a^x g(t)\,dt - \widetilde{G}_g(a, x, n)\right) \\ & - \left(\int_a^x \frac{Dt^{2n+3}}{(2n+3)!}dt - G\left[\frac{Dx^{2n+3}}{(2n+3)!}, 1, a, x, n+1\right]\right). \end{aligned}$$

The first term in parentheses on the RHS is the error term that we obtain when we apply our algorithm to $\int_a^x g(t)\,dt$, and the second term in parentheses is easy to compute exactly. The net result is the error term corresponding to $f(x)$, as desired. A similar process is used on the interval $[x, b]$.

3. An easy way to implement an RK method

$$\begin{aligned} y_{i+1} &= y_i + h_{i+1}F(x_i, y_i, h_{i+1}) \\ h_{i+1} &= x_{i+1} - x_i \end{aligned} \tag{6}$$

in the negative x direction, starting at a point x_0, is to simply label the nodes according to

$$\{\ldots, x_4, x_3, x_2, x_1, x_0\},$$

where we have $x_0 > x_1 > x_2 > x_3 > x_4 > \ldots$. Note that, in (6), $h_{i+1} < 0$ since $x_{i+1} < x_i$. There are two points that must be considered regarding the backward RK computation: (i) we must use $\widetilde{G}_f(x, b, n)$ or $\widetilde{G}_g(x, b, n)$ for the GL quadrature term and, (ii), we must use $-f(x)$ and $-g(x)$ in (3) and (4), respectively, due to x being the lower limit.

4. Assume $a \leqslant z < x$. We use the symbol $\widetilde{\Delta}_f(\alpha, \beta, n)$ to denote the error in the GL quadrature $\widetilde{G}_f(\alpha, \beta, n)$. We have

$$\int_a^b f(t)\,dt = \int_a^z f(t)\,dt + \int_z^b f(t)\,dt$$

and

$$\int_a^b f(t)\,dt = \widetilde{G}_f(a, b, n) + \widetilde{\Delta}_f(a, b, n)$$
$$\int_a^z f(t)\,dt = \widetilde{G}_f(a, z, n) + \widetilde{\Delta}_f(a, z, n)$$
$$\int_z^b f(t)\,dt = \widetilde{G}_f(z, b, n) + \widetilde{\Delta}_f(z, b, n).$$

Hence,

$$\widetilde{\Delta}_f(a, z, n) = \widetilde{G}_f(a, b, n) + \widetilde{\Delta}_f(a, b, n) - \widetilde{G}_f(a, z, n) - \widetilde{G}_f(z, b, n) - \widetilde{\Delta}_f(z, b, n)$$

which is easily determined since $\widetilde{\Delta}_f(a, b, n)$ has been found via the forward RK computation, $\widetilde{\Delta}_f(z, b, n)$ has been found via the backward RK computation, and the other three terms are quadrature terms determined by direct substitution of relevant known values. This process can also be applied when $g(x)$, instead of $f(x)$, has been used - simply replace the subscript f with g in the above - to obtain $\widetilde{\Delta}_g(a, z, n)$.

5. Bonnet's recursive formula for the Legendre polynomials on $[-1, 1]$ is

$$P_{n+1}(x) = \frac{(2n+1)\,xP_n(x) - nP_{n-1}(x)}{n+1}$$

with $P_0(x) = 1$ and $P_1(x) = x$.

References

[1] J.S.C. Prentice, *Estimating the error term in the Trapezium Rule using a Runge-Kutta method*, arXiv (Cornell University) (2023), DOI: https://doi.org/10.48550/arXiv.2306.06785

[2] D. Kincaid, W. Cheney, *Numerical Analysis: Mathematics of Scientific Computing, 3rd ed*, Brooks/Cole, Pacific Grove, 2002, ISBN: 0534389058.

[3] E. Isaacson, H.B. Keller, *Analysis of Numerical Methods*, Dover, New York, 1994, ISBN: 0486680290.

[4] J.S.C. Prentice, *The nature of the nodes, weights and degree of precision in Gaussian quadrature rules*, International Journal of Mathematical Education in Science and Technology, **42**, 1 (2010) 109-117, DOI: https://doi.org/10.1080/0020739X.2010.510222

[5] A.N. Lowan, N. Davids, A. Levenson, *Table of the zeros of the Legendre polynomials of order* $1-16$ *and the weight coefficients for Gauss' mechanical quadrature formula*, Bull. Amer. Math. Soc, **48** (1942) 739-743, DOI: https://doi.org/10.1090/S0002-9904-1942-07771-8

[6] J.C. Butcher, *Numerical Methods for Ordinary Differential Equations*. Wiley, Chichester, 2003, ISBN: 0471967580.

[7] E. Hairer, S.P. Norsett, G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Springer, Berlin, 2000, ISBN: 3540566708.

[8] J.S.C. Prentice, *Stepwise global error control in Euler's method using the DP853 triple and the Taylor remainder term*, arXiv (Cornell University Library) (2023), DOI: http://dx.doi.org/10.48550/arXiv.2303.09613

[9] J.S.C. Prentice, Relative Global Error Control in the RKQ Algorithm for Systems of Ordinary Differential Equations, *Journal of Mathematics Research*, **3**, 4 (2011) 59-66, DOI: https://doi.org/10.5539/jmr.v3n4p59

[10] J.S.C. Prentice, *Stepwise Global Error Control in an Explicit Runge-Kutta Method Using Local Extrapolation with High-Order Selective Quenching*, Journal of Mathematics Research, **3**, 2 (2011) 126-136, DOI: https://doi.org/10.5539/jmr.v3n2p126

Figures

For the convenience of the reader, all figures have been collected here.

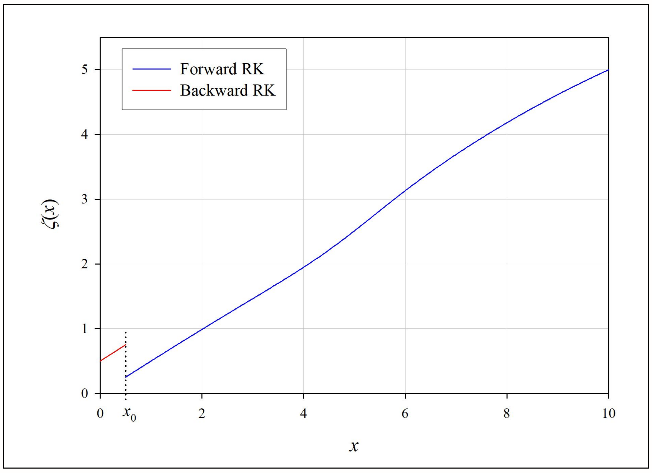

Figure 1. $\zeta(x)$ for the forward and backward RK computations, for $f(x) = x^2 \sin x$. The point x_0 is indicated.

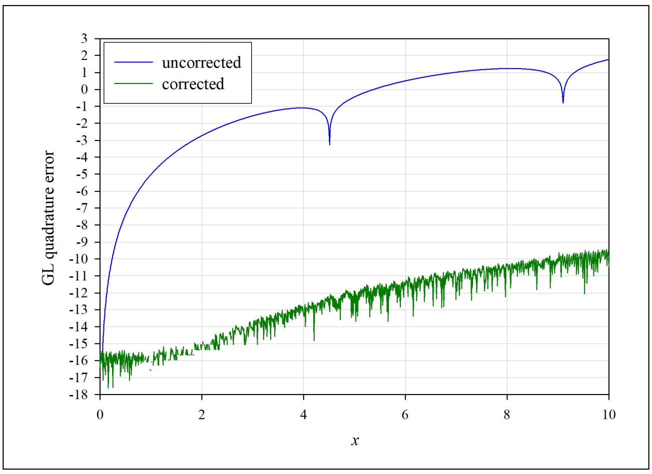

Figure 2. Quadrature error (blue) and corrected quadrature error (green), for $f(x) = x^2 \sin x$. Vertical axis is base-10 logarithmic.

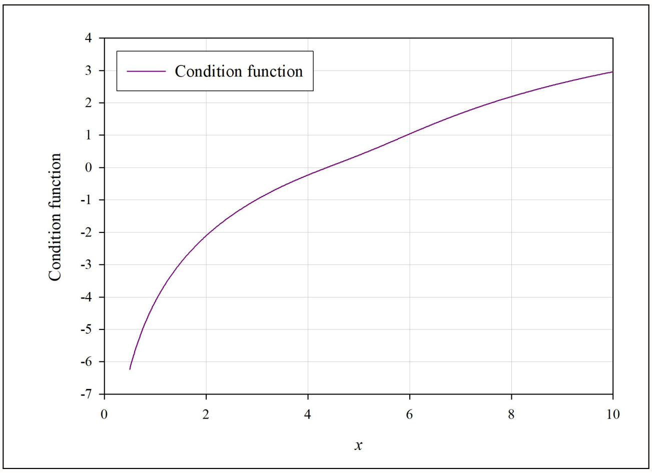

Figure 3. Condition function on $[x_0, x_N]$ for $f(x) = x^2 \sin x$. Vertical axis is base-10 logarithmic. The condition function clearly increases in magnitude quite significantly with x.